INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 10(3):390-402, June, 2015.

A Reference Dataset for Network Traffic Activity Based Intrusion Detection System

R. Singh, H. Kumar, R.K. Singla

Raman Singh*, Harish Kumar and R.K. Singla
University Institute of Engineering and Technology
Panjab University, Chandigarh, India
raman.singh@ieee.org, harishk@pu.ac.in, rksingla@pu.ac.in
*Corresponding author: raman.singh@ieee.org

Abstract: The network traffic dataset is a crucial part of anomaly based intrusion detection systems (IDSs). These IDSs train themselves to learn normal and anomalous activities. Properly labeled dataset is used for the training purpose. For the activities based IDSs, proper network traffic activity labeled dataset is the first requirement, however non-availability of such datasets is bottlenecked in the field of IDS research. In this experiment, a synthetic dataset "Panjab University - Intrusion Dataset (PU-IDS)" is created. The purpose of this study is to provide the researchers a reference dataset for the performance evaluation of network traffic activity based IDSs. University of New Brunswick Network Security Laboratory - Knowledge Disscovery in Databases (NSL-KDD) is a benchmark dataset for anomaly detection but it does not contain activity based labeling. So basic characteristics of this dataset are taken for the generation of the new synthetic dataset with various activities based labels. The dataset is first categorized as per *protocol* and *service*. Thereafter, as per minimum & maximum values of attributes, activity profiles are synthetically generated. This paper also discusses various statistical characteristics of PU-IDS. The total number of 198533 instances along with 273 of activity profiles are created. This dataset also contain different 98 *protocol_ service* profiles.
Keywords: Intrusion Detection System, Network Traffic Dataset, Network Traffic Profiling, Behavioral Profiling, Traffic Activity profiling

1 Introduction

Anomaly based Intrusion Detection System (IDS) is an emerging research interest for network security researchers and professionals. In order to detect intrusions, IDSs learn the network traffic activities/ behaviors of malware rather than rely on virus definitions and security updates. Well labeled network traffic dataset is used for this training. This dataset is crucial and only properly labeled dataset can serve this purpose. Profiling based anomaly detection is a new and emerging field of network security research, but, no standard network traffic activities based labeled dataset is available for the training and performance testing.
Some available datasets are internet traces and un-labeled while others are only single level labeled (normal or anomalous). Despite the wide availability of these datasets, non-availability of network traffic activities based datasets is hindrance in the intrusion detection research. In the anomaly based IDSs, normal behavior of networks may differ for different kind of organizations. For example, network game packets (online or intranet games) may be malicious for consultancy companies while they are absolutely normal for game development and testing companies. In the same manner within an organization, different departments may have different network traffic patterns. There is need of development of such anomaly detection techniques which can identify such network traffic activities/behaviors and create normal (and/or anomalous) profiles for intrusions detection. No such dataset exists for this purpose. The synthetic dataset should be generated so as to be used as a reference dataset for testing and performance evaluation. In this paper, University of New Brunswick Network Security Laboratory - Knowledge Disscovery

in Databases (NSL KDD) network traffic dataset is taken as a base and "Panjab University - Intrusion Dataset (PU-IDS)" network traffic dataset is generated synthetically. NSL KDD is benchmark dataset but does not contains network traffic activities based labels. These labels are introduces in PU-IDS.
This paper is divided into five Sections; Section 1 introduces the topic, Section 2 discusses the various available network traffic datasets. This Section also discusses available network traffic profiling based threat detection techniques. Section 3 describes the dataset used. This section also explain the methodology of synthetic dataset generation. In Section 4, statistical analysis of synthetic dataset is carried out. Finally, Section 5 discusses the conclusions and future works.

2 Network Traffic Dataset and Profiling : Related Work

2.1 Network Traffic Dataset

In the last few years, some datasets are either generated or collected for research purpose. These datasets are used by researchers worldwide. The Cyber Systems and Technology Group (formerly the DARPA Intrusion Detection Evaluation Group) of MIT Lincoln Laboratory has collected network traffic dataset in 1998 and 1999. This dataset is collected by simulating various attacks like denial of services (DOS), remote to local (R2L), user to remote (U2R) etc. on different platforms like Windows, Unix etc. This dataset become benchmark KDD dataset for research in IDS [1]. Later, some of the problems like redundant and duplicate records present in this dataset are removed and more efficient dataset become available for researcher which is known as NSL KDD dataset [2]. This dataset is widely used for performance analysis of various intrusion detection techniques. In this dataset each instances are labeled as normal or anomalous. In another version of this dataset, instances are classified as par various attacks like dos,U2R, L2R, probe and others. The drawback of this dataset is that, no further network traffic activities/behaviors based labels are available. The Stanford Network Analysis Project created 50 network traffic datasets by using various nodes of social networks, internet work, web graphs, etc. Some of this collection of datasets is labeled while others are un-labeled. Profiling based classification is missing in this dataset [3].
The dataset is created by the Center for Applied Internet Data Analysis (CAIDA) at various topologically and geographically separated locations. Anonymized internet traces along with other worms related dataset is created by ensuring privacy preservation from the year 2008 to the year 2013. This is a collection of various datasets like anonymized internet traces and attack specific dataset. The internet traces dataset is un-labeled and so may not be useful directly for performance analysis of IDS. It needs to be the first pre-processed for labeling [4]. CERT synthetic *sendmail* system dataset is created by the University of New Mexico using Sun SPARC stations. Some *system call* instances are live while others are synthetic. Normal instances and intrusion traces are provided separately. Further level of labeling is not provided in this dataset [5]. Real traffic of HTTP, SMTP, SSH, IMAP, POP3, and FTP are used to create a network traffic dataset in the University of New Brunswick Information Security ? Centre of Excellence (UNB - ISCX) lab. This dataset which contains seven days of normal and malicious instances is known as UNB ISCX dataset [6]. This dataset is available on request for university researchers. This dataset includes normal and malicious instances. Profiling level labeling is not available for this dataset. Internet traces of the sub-network of Panjab University network (PU-CAN) is captured from July 2011 to August 2011 in order to create a network traffic dataset. This dataset consists of un-labeled internet traces which needs further processing for labeling [7]. Internet traffic traces like LAN/WAN Ethernet traffic, TCP traffic, HTTP traffic, HTTP logs etc., are available in other dataset. The collection of datasets includes LAN/WAN traces, various

logs and web client traces. A further level of classification and labeling is not present in these datasets [8].
From the literature studied it has been found that most of the datasets are not labeled on the basis of network traffic activities/behaviors. These datasets are labeled on single level profiling. Therefore, there is a need for the development of a dataset which is based on the behavior profiling that consider network traffic activities and various user's behaviors.

2.2 Network Traffic Profiling

Researchers are using clustering and profiling in order to detect intrusions. Network traffic activities based IDSs are the future of network security. In this sub-section various techniques which are used to detect intrusions are discussed. Clustering is used in the network traffic dataset in order to cluster various instances into different activity profiles. These profiles are used to determine normal and malicious instances [9]. Behavioral distance based anomaly detection for real time traffic analysis is also proposed. Horizontal and vertical distance metrics are used for different attributes of network traffic datasets [10]. Behavioral foot printing method along with content based signature is used to profile self-propagating worms. The worm's dynamic infection sequence is learned to detect it [11].
Clustering based anomaly detection technique is proposed for modeling user's normal behavior in [12]. Traffic causality graphs (TCGs) are used to analyze and visualize temporal and spatial causality of flows to profile network traffic. The advantage of this technique is that there is no need of payload inspection [13]. Personal and application profiles are created by tagging network traffic. Traffic from known source is profiled with role tag and application tag [14]. IP to IP communication graph and information is used to profile internet backbone traffic. This technique is known as profiling by association [15]. Data mining and entropy based techniques are used to profile the behavior of internet end to end hosts [16]. Researchers also proposed behavior based tracking to create long term profiles of user?s interest. This methodology is tested on DNS traffic [17].
These techniques can be used on network traffic activity based datasets in order to detect various normal and anomalous network traffic activities/behaviors. In the state of the art IDSs, the normal behavior of user's groups needs to be learned to effectively detect intrusions.

3 Materials and Methods

Unavailability of traffic activity/ behavior level network traffic dataset motivates this experiment of generation of synthetic dataset. The network traffic dataset used for generation of the synthetic dataset is discussed as below:

3.1 Network Traffic Dataset Used

Benchmarked NSL-KDD network traffic dataset is used as a base and its various characteristics of different attributes are used to synthetically generate instances. The various characteristics used are minimum and maximum values of different attributes. Table 1 shows the list of attributes of this dataset. The same attributes are taken in PU-IDS. These attributes are either continuous or categorical type. Continuous attributes are those in which the value belongs to indefinite set while in categorical attribute values are assigned from a definite set. Attribute number 2, 3, 4, 7, 12, 14, 15, 21, 22 and 42 are categorical attributes while all others are continuous attributes.

Table 1: List of Attributes of NSL KDD and Synthetically generated dataset

Sr. No.	Feature	Sr. No.	Feature	Sr. No.	Feature
1	Duration	15	Su attempted	29	Same srv rate
2	Protocol type	16	Num root	30	Diff srv rate
3	Service	17	Num file creations	31	Srv diff host rate
4	Flag	18	Num shells	32	Dst host count
5	Source bytes	19	Num access files	33	Dst host srv count
6	Destination bytes	20	Num outbound cmds	34	Dst host same srv rate
7	Land	21	Is host login	35	Dst host diff srv rate
8	Wrong fragment	22	Is guest login	36	Dst host same src port rate
9	Urgent	23	count	37	Dst host srv diff host rate
10	Hot	24	Srv count	38	Dst host serror rate
11	Number failed logins	25	Serror rate	39	Dst host srvserror rate
12	Logged in	26	Srvserror rate	40	Dst host rerror rate
13	Num compromised	27	Rerror rate	41	Dst host srvrerror rate
14	Root shell	28	Srvrerror rate	42	Class label

3.2 Synthetic Dataset Generation Methodology

Figure 1 shows the methodology of synthetic dataset generation. It describes the procedures followed in the experiment to generate instances of synthetic dataset. This methodology is implemented using Matlab [18] as a tool. The various steps are described as below:

Separation of dataset

In the first step, NSL KDD training and testing dataset is clubbed into a single set. This dataset is then separated into two categories of normal and anomalous sub-datasets.

Protocol_ service profile creation

In the second step, *protocol_ service* profiles are created by integrating protocol and service attributes for both categories of sub-datasets (normal and anomalous sub-datasets).

All *protocol_ service* profiles created in the first level of profiling are shown in table 2. Thereafter, all instances are separated as per *protocol_ service* profiles for both normal and anomalous sub-datasets.

Extraction of basic characteristics

In the third step, all continuous attributes of all *protocol_ service* profiles of both sub-datasets are considered and basic characteristics like minimum and maximum values are extracted. Unique values for each categorical attributes are also calculated.

Calculation of *cluster_ gap* and *number_ of_ instances_ per_ cluster*

In the fourth step, for each *protocol_ service* profile, number of clusters and the number of instances to be created are taken as input from the user.

As per equation 1 *cluster_ gap* and as per equation 2 *number_ of_ instances_ per_ cluster* is calculated for each *protocol_ service* profile.

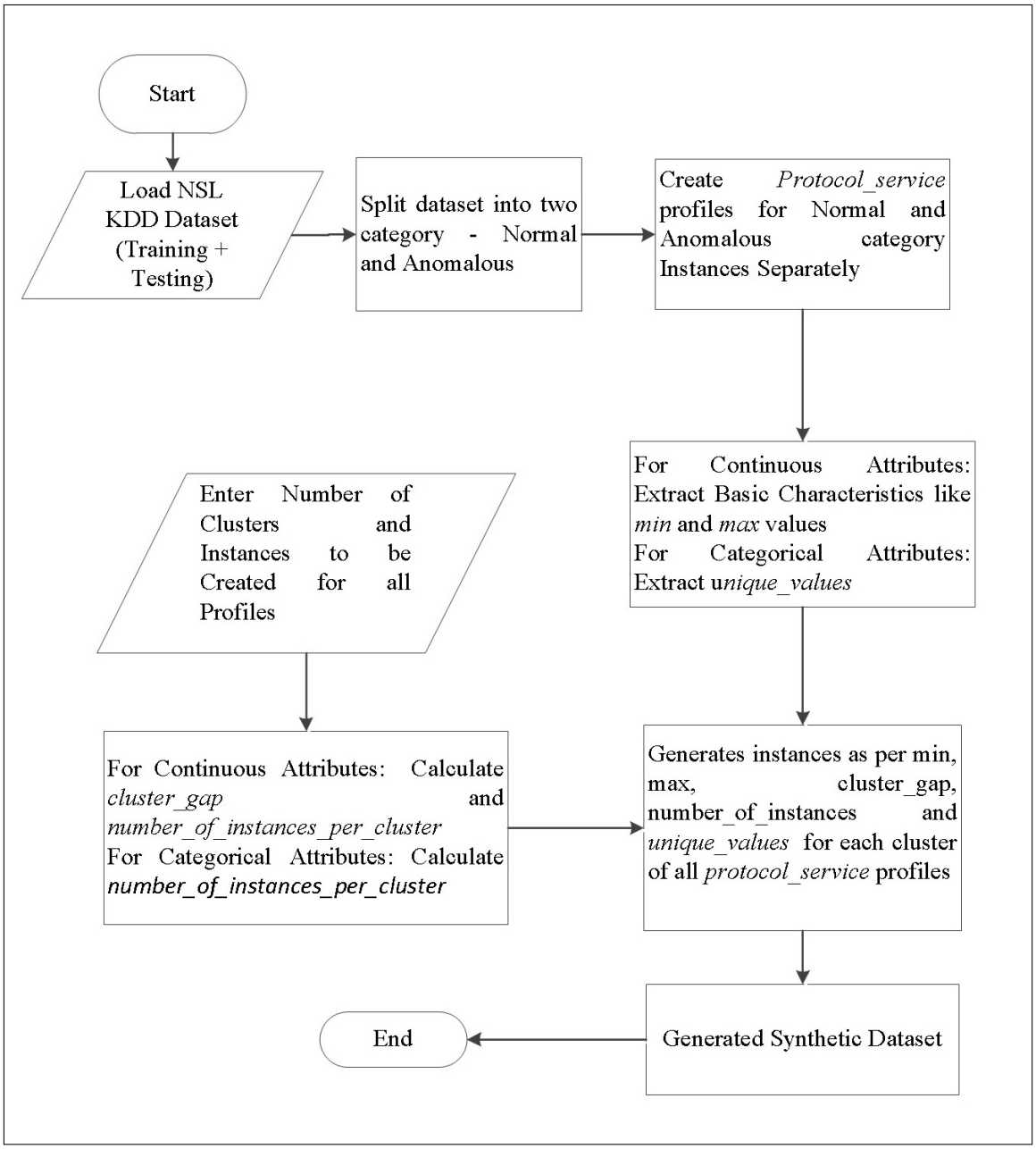

Figure 1: Synthetic dataset generation methodology

Table 2: List of *protocol_ service* Profiles

icmp_eco_i	tcp_courier	tcp_gopher	tcp_link	tcp_pop_2	tcp_systat
icmp_ecr_i	tcp_csnet_ns	tcp_harvest	tcp_login	tcp_pop_3	tcp_telnet
icmp_red_i	tcp_ctf	tcp_hostnames	tcp_mtp	tcp_printer	tcp_time
icmp_tim_i	tcp_daytime	tcp_http	tcp_name	tcp_private	tcp_uucp
icmp_urh_i	tcp_discard	tcp_http_2784	tcp_netbios_dgm	tcp_remote_job	tcp_uucp_path
icmp_urp_i	tcp_domain	tcp_http_443	tcp_netbios_ns	tcp_rje	tcp_vmnet
tcp_IRC	tcp_echo	tcp_http_8001	tcp_netbios_ssn	tcp_shell	tcp_whois
tcp_X11	tcp_efs	tcp_imap4	tcp_netstat	tcp_smtp	udp_domain_u
tcp_Z39_50	tcp_exec	tcp_iso_tsap	tcp_nnsp	tcp_sql_net	udp_ntp_u
tcp_aol	tcp_finger	tcp_klogin	tcp_nntp	tcp_ssh	udp_other
tcp_auth	tcp_ftp	tcp_kshell	tcp_other	tcp_sunrpc	udp_private
tcp_bgp	tcp_ftp_data	tcp_ldap	tcp_pm_dump	tcp_supdup	udp_tftp_u

In these equations x_{max} is the maximum and x_{min} is the minimum value extracted in step three for a particular *protocol_ service* profile. $k_{cluster}$ is the number of clusters required and *number_ of_ instances_ protocol_ service* is the number of instances required for particular *protocol_ service* profile which are taken from the user.

$$cluster\_gap = \frac{x_{max} - x_{min}}{k_{cluster}} \tag{1}$$

$$number\_of\_instances\_per\_cluster = round\{\frac{number\_of\_instances\_protocol\_service}{k_{cluster}}\} \tag{2}$$

Table 3 shows various parameters like $k_{cluster}$ and *number_ of_ instances_ protocol_ service* used to generate synthetic dataset.

In table 3, *Cluster_ N* ($k_{cluster}$) and *Insta_ N* (*number_ of_ instances_ protocol_ service*) corresponds to the number of clusters and instances to be created for normal dataset respectively.

Cluster_ A and *Insta_ A* is the number of clusters and instances to be created for anomalous dataset respectively.

Table 3: Synthetic Dataset Generation Parameters

Protocol Service Profiles	Cluster _N	Insta _N	Cluster _A	Insta _A	Protocol Service Profiles	Cluster _N	Insta _N	Cluster _A	Insta _A
icmp_eco_i	3	1215	6	4200	tcp_link	1	5	4	1536
icmp_ecr_i	2	456	4	3120	tcp_login	0	0	3	855
icmp_red_i	1	125	0	0	tcp_mtp	0	0	3	858
icmp_tim_i	1	220	1	42	tcp_name	0	0	4	780
icmp_urh_i	1	150	0	0	tcp_netbios_dgm	0	0	4	2920
icmp_urp_i	4	1200	1	45	tcp_netbios_ns	0	0	3	960
tcp_IRC	3	900	1	50	tcp_netbios_ssn	0	0	2	1048
tcp_X11	1	175	1	124	tcp_netstat	0	0	3	1305
tcp_Z39_50	0	0	5	2800	tcp_nnsp	0	0	4	2600
tcp_aol	0	0	1	50	tcp_nntp	0	0	2	708
tcp_auth	2	450	4	952	tcp_other	2	460	5	1850
tcp_bgp	0	0	6	2100	tcp_pm_dump	0	0	1	135
tcp_courier	0	0	7	2100	tcp_pop_2	0	0	1	235
tcp_csnet_ns	0	0	5	3270	tcp_pop_3	1	350	1	115
tcp_ctf	0	0	4	800	tcp_printer	0	0	1	354
tcp_daytime	0	0	6	1800	tcp_private	1	22	6	28000
tcp_discard	0	0	4	1680	tcp_remote_job	1	5	1	154
tcp_domain	1	255	3	405	tcp_rje	0	0	1	204
tcp_echo	0	0	4	1040	tcp_shell	1	16	1	165
tcp_efs	0	0	5	1850	tcp_smtp	5	9600	2	386
tcp_exec	0	0	3	1395	tcp_sql_net	0	0	2	840
tcp_finger	3	375	5	1750	tcp_ssh	1	20	2	400
tcp_ftp	4	1616	4	1800	tcp_sunrpc	0	0	2	856
tcp_ftp_data	6	3006	5	2000	tcp_supdup	0	0	3	1404
tcp_gopher	0	0	3	570	tcp_systat	0	0	2	1206
tcp_harvest	0	0	1	26	tcp_telnet	5	5750	3	1836
tcp_hostnames	0	0	2	500	tcp_time	1	170	2	680
tcp_http	6	50010	4	2800	tcp_uucp	0	0	3	2085
tcp_http_2784	0	0	1	12	tcp_uucp_path	0	0	3	2346
tcp_http_443	0	0	3	999	tcp_vmnet	0	0	3	1875
tcp_http_8001	0	0	1	14	tcp_whois	0	0	3	2256
tcp_imap4	1	160	3	888	udp_domain_u	5	11825	1	14
tcp_iso_tsap	0	0	4	820	udp_ntp_u	1	320	0	0
tcp_klogin	0	0	3	924	udp_other	4	2624	2	304
tcp_kshell	0	0	2	650	udp_private	3	1212	4	2600
tcp_ldap	0	0	2	360	udp_tftp_u	1	35	0	0

Synthetic instances generation

In the fifth step, for each continuous attributes of various *protocol_service* profiles, $k_{cluster}$ numbers of traffic activity/ behavior profiles are created by using x_{min}, x_{max} and $k_{cluster}$.

For categorical attributes of various *protocol_service* profiles $k_{cluster}$ number of traffic activity/ behaviors are created by considering *cate_unique_values*.

For each traffic activity profile, *number_of_instances_per_cluster* numbers of instances are

generated by using same characteristics used for generation of various traffic activity/ behavior profiles.

Values of continuous attributes for each cluster is calculated as given in equation (3)

$$value\_conti\_Attribute = max\_value\_previous\_cluster + cluster\_gap \tag{3}$$

Subject to:
$max\_value\_previous\_cluster = x_{min}$, for first cluster
and
$x_{min} \geq value\_conti\_attribute \leq x_{max}$ for other clusters

Where *value_conti_attribute* represents value of a particular continuous attribute and *max_value_previous_cluster* represents the maximum value of previous cluster of that particular attribute.

For categorical attributes *cate_unique_values* are extracted by taking each unique value for each categorical attribute separately as shown in equation (4).

$$value\_cate\_Attribute = div\{cate\_unique\_values(k_{\text{cluster}})\} \tag{4}$$

Function *div* means values of particular categorical attributes are obtained by dividing *cate_unique_value* into cluster groups and then for each group/ cluster *number_of_instances_per_cluster* instances are created.

3.3 Illustrative Example

let?s say for *tcp_http protocol_service profile*, 3 clusters (k_{cluster}) and 1000 instances (*number_of_instances_protocol_service*) are required to be generated.

So the values of x_{min} and x_{max} for all continuous attributes are extracted. Also assume the values of x_{min} and x_{max} for attribute *source_bytes* are 30 and 255. So *cluster_gap* and *number_of_instances_per_cluster* are calculated as per equation (1) and (2) respectively as shown in example 1.

Example 1. *Cluster_gap = (255-30) /3 = 75*
number_of_instances_per_cluster = round (1000/3) = 333
Range of First Cluster value_conti_attribute1 = (from 30 up to 30+75=105)
Range of Second Cluster value_conti_attribute2 = (from 105 up to 105+75=180)
Range of Third Cluster value_conti_attribute3 = (from 180 up to x_{max} = 255)

For each continuous attribute , 333 instances per cluster are created and the same procedure is followed for all continuous attributes (Last cluster may have one additional instance to sum up total instances to 1000). Example 2 explain the assignment of values of different clusters for categorical attributes (like *flag*).

Example 2. *cate_unique_values = S0, S1 ,S2, S3, S4, S5, S6, S7, S8 (assumed value)*
For First Cluster, value_cate_attribute1 = S0, S1, S2
For Second Cluster, value_cate_attribute2= S3, S4, S5
For Third Cluster, value_cate_attribute3= S6, S7, S8

Now for each categorical attribute, 333 instances per cluster are created by taking value from *value_cate_attribute* for all categorical attributes (Last cluster may have one additional instance to sum up total instances to 1000). These procedures are followed for each continuous

and categorical attributes of all *protocol_ service* profiles and for both categories of sub-datasets. Both categories of normal and anomalous synthetic sub-datasets are integrated to obtain one synthetically generated dataset. This dataset has two levels of profile labeling. One level is *protocol_ service* while other is *traffic activity/ behavior* level.

4 Results And Analysis

The NSL KDD (Training + Testing) dataset taken for experimentation has 148517 instances and 72 *protocol_ service* profiles. The same numbers of protocol *protocol_ service* are created in synthetic dataset. Table 4 shows the comparative statistical analysis of NSL KDD and PU-IDS dataset.

Table 4: Basic statistics of NSL KDD and PU-IDS dataset

Statistics	NSL KDD Dataset		PU-IDS	
	Normal	Anomaly	Normal	Anomaly
Numbers of instances	77054	71463	92727	105806
Numbers of *protocol_ service* profiles	30	68	30	68
Numbers of network traffic activity profiles	Not present	Not present	72	201

In NSL KDD dataset, 30 normal and 68 anomalous *protocol_ service* profiles are present. Some of these profiles are common in both normal and anomalous sub-datasets. Total 198533 instances are generated synthetically out of which 92727 are normal while other are anomalous. In PU-IDS, the generated instances are more than the base dataset. These excess instances provide an opportunity of better and effective training of IDS as dimensions of normal and anomalous behavior is increased. These instances also helps in performance testing of IDS in broad spectrum. Equal numbers of *protocol_ service* profiles as of base dataset are created in PU-IDS dataset. Generated dataset has equal first level of profiles. In NSL KDD dataset, traffic activity labeling are not present and hence this second level of labeling is synthetically generated. Different 273 traffic activities in datasets are synthetically generated as shown in table 4. 72 normal and 201 anomalous behavior profiles are generated. This second level of labeling will help the researcher to train and test anomaly detection techniques, where NSL KDD provides limited opportunity. Figure 2 shows the number of normal instances generated for each *protocol_ service* profile.

53.93% of the generated normal instances are of *tcp_ http protocol_ service*. This profile is the biggest contributor in normal instances. The different significant normal *protocol_ service* profiles are *udp_ domain_ u* (12.75%), *tcp_ smtp* (10.35%), *tcp_ telnet* (6.2%), *tcp_ ftp_ data* (3.24%), *udp_ other* (2.83%), *tcp_ ftp* (1.74%), *icmp_ eco_ i* (1.31%), *udp_ private* (1.31%) and *icmp_ urp_ i* (1.29%). Other *protocol_ service* (*tcp_ IRC, tcp_ other, icmp_ ecr_ i, tcp_ auth, tcp_ finger, tcp_ pop_ 3, udp_ ntp_ u, tcp_ domain, icmp_ tim_ i, tcp_ X11, tcp_ time, tcp_ imap4, icmp_ urh_ i, icmp_ red_ i, udp_ tftp_ u, tcp_ private, tcp_ ssh, tcp_ shell, tcp_ link, and tcp_ remote_ job*) collectively contribute 5.04% of total normal instances. Only 0.0053% of *tcp_ remote_ job* instances are generated which is lowest in normal instances.

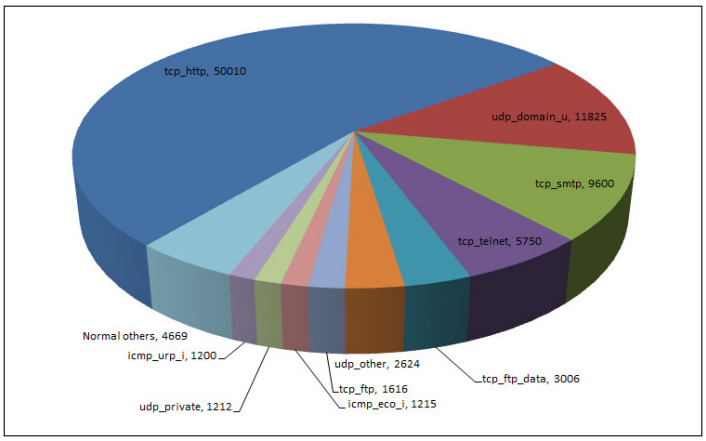

Figure 2: Distribution of normal instances as per *protocol_ service* profiles

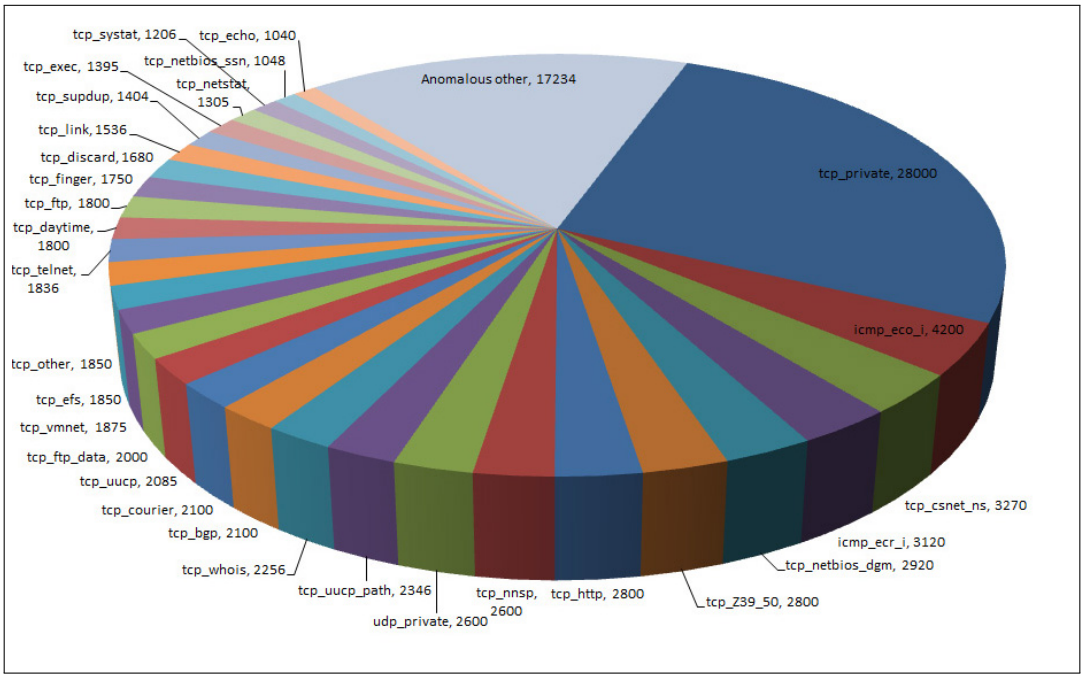

Figure 3: Distribution of anomalous instances as per *protocol_ service* profiles

Figure 3 shows the number of anomalous instances generated for each *protocol_service* profile. In the anomalous instances the *tcp_private* profile has largest share, which is 26.46%. The various significant anomalous *protocol_service* profiles are *icmp_eco_i* (3.97%), *tcp_csnet_ns* (3.09%), *icmp_ecr_i* (2.95%) and *tcp_netbios_dgm* (2.76%). Other anomalous *protocol_service* profiles (like *tcp_http_443, tcp_netbios_ns, tcp_auth, tcp_klogin, tcp_imap4, tcp_mtp, tcp_sunrpc, tcp_login, tcp_sql_net, tcp_iso_tsap, tcp_ctf, tcp_name, tcp_nntp, tcp_time, tcp_kshell, tcp_gopher, tcp_hostnames, tcp_domain, tcp_ssh, tcp_smtp, tcp_ldap, tcp_printer, udp_other, tcp_pop_2, tcp_rje, tcp_shell, tcp_remote_job, tcp_pm_dump, tcp_X11, tcp_pop_3, tcp_IRC, tcp_aol, icmp_urp_i, icmp_tim_i, tcp_harvest, tcp_http_8001, udp_domain_u, and tcp_http_2784*) contribute 16.29% of total anomalous instances. Least number of anomalous instances of *tcp_http_2784 protocol_service* (only 0.011%) are generated.
Figure 4 shows the comparative study of various protocols like *tcp*, *udp* and *icmp*, present in NSL-KDD and synthetically generated PU-IDS dataset. In the base dataset 121569 of *tcp*, 17614

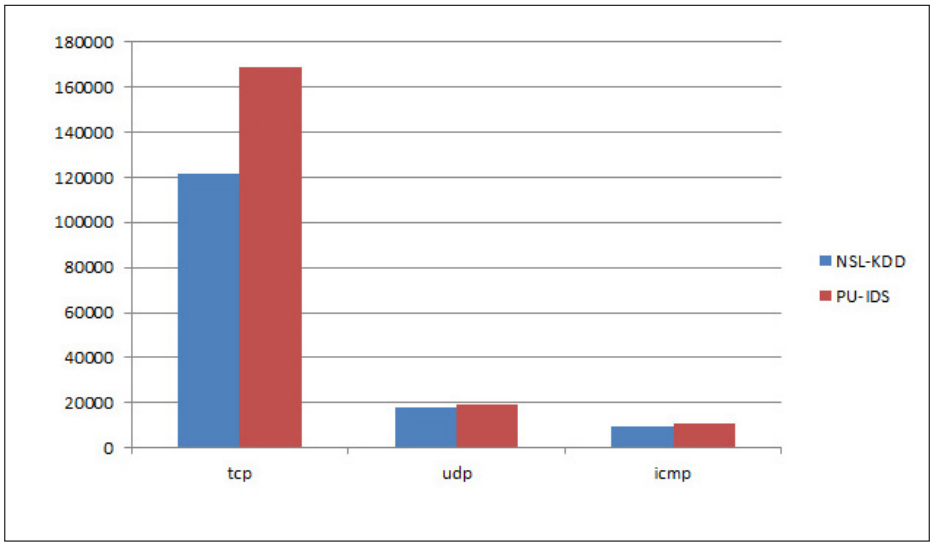

Figure 4: Numbers of instances of *tcp*, *udp* and *icmp* protocols

of *udp* and 9334 of *icmp* instances are present. In PU-IDS, the numbers of synthetically generated instances of *tcp*, *udp* and *icmp* are 168826, 18934 and 10773 respectively.

5 Conclusions and Future Works

The network traffic dataset is an important part of Intrusion Detection System as it learns normal and anomalous behavior of computer networks. Perfect training of IDSs depends on the proper labeled dataset. The well-known benchmark NSL KDD dataset is widely used in IDS research, but this dataset is very old and only has one level of labeling. The other available unlabeled datasets has little importance in performance evaluation of malware detection techniques. Traffic activity labeled datasets are not present for research purpose. Activity/ behavioral based network traffic datasets are needed for state of the art IDSs. A new synthetic network traffic dataset (PU-IDS) with two levels of labeling is generated by taking basic characteristics of the NSL KDD dataset. "Protocol service" level and "traffic activity/ behavior" level labeling is created so as this dataset can be used for performance evaluation. This will overcome limited utility of NSL KDD network traffic dataset. PU-IDS consists of total numbers of 198533 instances along

with 72 protocol profiles. It also consists 273 synthetically created traffic activity/ behavioral profiles which is the novelty of this dataset. In the future, an organizational network with different departments and various user groups should be set up to create the simulated dataset. Users within one organization should exhibit similar network activities/ behaviors as they are working on the same set of software. Behavior based IDSs should identify these similar network activities and create normal or anomalous model.

Acknowledgment

This research work is supported by World Bank funded Technical Education Quality Improvement Program ? Phase II (TEQIP - II) in the form of research assistantship. This research is revised and extended version of paper presented at International Conference on Computers, Communications and Control (ICCCC 2014) held at Oradea, Romania on May 7-9, 2014.

Bibliography

[1] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[2] http://nsl.cs.unb.ca/NSL-KDD

[3] http://snap.stanford.edu/data

[4] http://www.caida.org/data/overview

[5] http://www.cs.unm.edu/ immsec/data

[6] http://www.iscx.ca/datasets

[7] Singh, R., Kumar H., Singla R.K (2012); Traffic Analysis of Campus Network for Classification of Broadcast Data. *47th Annual National Convention of Computer Society of India. Int. Conf. on Intelligent Infrastructure*, MacGraw Hill Professional: 163-166.

[8] http://ita.ee.lbl.gov/html/traces.html

[9] Marchette, D. (1999); A Statistical Method for Profiling Network Traffic, *Workshop on Intrusion Detection and Network Monitoring* : 119-128.

[10] Sengar, H.; Wang, X.; Wang, H.; Wijesekera, D.; Jajodia, S. (2009); Online detection of network traffic anomalies using behavioral distance, *17th Int. Workshop on Quality of Service* : 1-9.

[11] Jiang, X.; Zhu X. vEye (2009); Behavioral footprinting for self-propagating worm detection and profiling, *Knowledge and information systems* ; 18(2): 231-262

[12] Oh, H.S.; Lee, W.S. (2003); An anomaly intrusion detection method by clustering normal user behavior, *Computers & Security*, 22(7): 596-612.

[13] Asai, H.; Fukuda, K. ; Esaki, H. (2011); Traffic causality graphs: profiling network applications through temporal and spatial causality of flows, *Proc. of the 23rd Int. Teletraffic Congress* : 95-102.

[14] Zoquete, A.; Correia, P.; Shamalizadeh, H. (2011); Packet tagging system for enhanced traffic profiling. *IEEE 5th Int. Conf. on Internet Multimedia Systems Architecture and Application (IMSAA)* : 1-6.

[15] Iliofotou, M.; Gallagher, B.; Eliassi-Rad, T.; Xie, G.; Faloutsos, M.(2010); Profiling-by-association: a resilient traffic profiling solution for the internet backbone. *Proc. of the 6th Int. Conference Co-NEXT'10* : DOI: 10.1145/1921168.1921171.

[16] Xu, K.; Zhang, Z.L.; Bhattacharyya S.(2008); Internet traffic behavior profiling for network security monitoring. *IEEE/ACM Trans. on Networking*, 16(6): 1241-1252.

[17] Herrmann, D.; Banse, C.; Federrath, H.(2013); Behavior-based tracking: Exploiting characteristic patterns in DNS traffic. *Computers & Security*, 39 (Part A): 17-33.

[18] http://www.mathworks.in/products/matlab